

IN THE CLAIMS

Please amend the claims to read as follows:

1. (Currently Amended) A computer, comprising:
a memory; and
a data structure for a generic document, the data structure stored in the memory, the data structure memory and including:
a definition of a first element, the definition of the first element including an element value field;
a key identifier to identify a key value field to be used as a key in a data store.
2. (Original) A computer according to claim 1, wherein:
the definition of the first element includes the key identifier as a property of the first element;
and
the key identifier identifies the element value field as the key value field.
3. (Original) A computer according to claim 1, wherein:
the definition of the first element includes a definition of a first attribute of the first element,
the definition of the first attribute including the key identifier; and
the key identifier identifies the element value field as the key value field.
4. (Original) A computer according to claim 1, wherein the definition of the first element includes a definition of a second attribute, the definition of the second attribute including an attribute value field.
5. (Original) A computer according to claim 4, wherein:
the definition of the second attribute includes the key identifier as a property of the second attribute; and
the key identifier identifies the attribute value field as the key value field.
6. (Original) A computer according to claim 4, wherein:
the definition of the first element further includes a definition of a third attribute of the first element, the definition of the third attribute including the key identifier; and

the key identifier identifies the attribute value field of the second attribute as the key value field.

7. (Original) A computer according to claim 1, further comprising:
a second element; and
a tree structure including the first element and the second element.

8. (Original) A computer according to claim 1, wherein the key identifier identifies the key value field as one of a primary key, a secondary key, and a foreign key for the data store.

9. (Original) A computer according to claim 8, wherein:
the key identifier identifies the key value field as a foreign key for the data store; and
the key identifier references a second data store.

10. (Currently Amended) A computer according to claim 1, wherein the ~~schema is data~~
structure contains an eXtensible Markup Language (XML) schema.

11. (Currently Amended) A computer system, comprising:
a data store to store a first generic document; and
a first schema applicable to the first generic document, the first schema including:
a definition of a first element, the definition of the first element including an element value field;
a first key identifier to identify a first key value field in the first generic document to be used as a key in a data store.

12. (Original) A system according to claim 11, wherein:
the definition of the first element includes the first key identifier as a property of the first element; and
the first key identifier identifies an element value field of the first element as the first key value field.

13. (Original) A system according to claim 11, wherein:

the definition of the first element includes a definition of a first attribute, the definition of the first attribute including the first key identifier; and

the first key identifier identifies an element value field of the first element as the first key value field.

14. (Original) A system according to claim 11, wherein the definition of the first element includes a definition of a second attribute, the definition of the second attribute including an attribute value field.

15. (Original) A system according to claim 14, wherein:
the definition of the second attribute includes the first key identifier as a property of the second attribute; and

the first key identifier identifies the attribute value field as the first key value field.

16. (Original) A system according to claim 14, wherein:
the definition of the first element further includes a definition of a third attribute, the definition of the third attribute including the first key identifier; and
the first key identifier identifies the attribute value field as the first key value field.

17. (Currently Amended) A system according to claim 11, wherein:
the data store is operative to store a second generic document; and
the system further comprises a second schema applicable to the second generic document, the second schema including:

a definition of a second element, the definition of the second element including an element value field;

a second key identifier to identify a second key value field in the second generic document to be used as a key in a data store.

18. (Original) A system according to claim 17, wherein:
the first schema includes a first identifier for the first key value field;
the second schema includes a second identifier for the second key value field; and
the first identifier and the second identifier are the same identifier.

19. (Original) A system according to claim 11, wherein the data store is a Lightweight Directory Access Protocol (LDAP) data store.

20. (Original) A system according to claim 11, further comprising a parser to parse the schema.

21. (Currently Amended) A system according to claim 20, further comprising:
the parser is operative to identify the first key value field in the schema; and
a loader to load a value from the first key value field in the first generic document.

22. (Currently Amended) A system according to claim 21, wherein the data store further includes an index associated with the first generic document, the index storing a copy of the value from the first key value field in the first generic document.

23. (Original) A system according to claim 22, wherein the index is in a native format of the data store.

24. (Currently Amended) A system according to claim 21, wherein:
the schema includes a definition of at least one of a second element and a fourth attribute;
the parser is operative to identify the second element or the fourth attribute;
the loader is operative to load a second value from the second element or the fourth attribute
in the first generic document; and
the data store further includes a field to store the second value in a native format of the data store.

25. (Original) A system according to claim 20, wherein:
the parser is operative to parse the schema into objects; and
the system further comprises a definer to define a structure for the data store based on the objects.

26. (Currently Amended) A system according to claim 11, wherein:
the first generic document is an eXtensible Markup Language (XML) document; and
the first schema is an XML schema.

27. (Currently Amended) A method for adding a generic document to a data store in a computer, comprising:

- accessing a schema for the generic document;
- locating a key value field in the generic document defined in the schema as a key;
- loading a value from the key value field;
- storing the value in the data store; and
- storing the generic document in the data store.

28. (Currently Amended) A method according to claim 27, further comprising indexing the generic document in the data store using the value.

29. (Original) A method according to claim 27, wherein identifying a key value field includes:

- locating an element defined in the schema as the key; and
- selecting an element value field for the element as the key value field.

30. (Original) A method according to claim 29, wherein locating an element includes locating the element based on a property of the element identifying the element as the key.

31. (Original) A method according to claim 29, wherein locating an element includes locating the element based on an attribute of the element identifying the element as the key.

32. (Original) A method according to claim 27, wherein identifying a key value field includes:

- locating a first attribute of an element defined in the schema as the key; and
- selecting an attribute value field for the first attribute as the key value field.

33. (Original) A method according to claim 32, wherein locating a first attribute includes locating the first attribute based on a property of the first attribute identifying the first attribute as the key.

34. (Original) A method according to claim 32, wherein locating an attribute includes locating the first attribute based on a second attribute of the element identifying the first attribute as the key.

35. (Currently Amended) A method according to claim 27, wherein storing the generic document includes:

parsing the generic document into objects according to the schema;
loading values for each object in the generic document; and
storing the values in the data store.

36. (Original) A method according to claim 35, wherein storing the values includes storing the values in the data store in a native format of the data store.

37. (Currently Amended) A method according to claim 27, wherein:
the generic document is an eXtensible Markup Language (XML) document; and
the schema is an XML schema.

38. (Currently Amended) A method for defining a data store in a computer, comprising:
accessing a schema;
locating an object defined in the schema as a key;
defining a first data structure in the data store for the object;
identifying the first data structure in the data store as a key data structure; and
defining a second data structure in the data store for a generic document conforming to the schema.

39. (Original) A method according to claim 38, wherein defining a first data structure includes defining the first data structure in the data store for the object in a native format of the data store.

40. (Original) A method according to claim 38, wherein defining a second data structure includes:
parsing the schema into objects; and
defining a data structure in the data store for each object.

41. (Original) A method according to claim 40, wherein defining a data structure in the data store for each object includes defining the data structure in the data store for each object in a native format of the data store.

42. (Currently Amended) A method according to claim 38, wherein:
the generic document is an eXtensible Markup Language (XML) document; and
the schema is an XML schema.

43. (Currently Amended) A method for defining a schema for a generic document in a computer, comprising:
defining a first element in the schema, the first element including an element value field; and
identifying a key value field in the schema to be used as a key in a data store.

44. (Original) A method according to claim 43, wherein:
defining a first element includes assigning a property to the first element as the key; and
identifying a key value field includes identifying the element value field as the key value field.

45. (Original) A method according to claim 43, wherein:
defining a first element includes defining a first attribute of the first element, the first attribute identifying the first element as the key; and
identifying a key value field includes identifying the element value field as the key value field.

46. (Original) A method according to claim 43, wherein defining a first element includes defining a second attribute for the first element, the second attribute including an attribute value field.

47. (Original) A method according to claim 46, wherein:
defining a second attribute includes assigning a property to the second attribute as the key;
and

identifying a key value field includes identifying the attribute value field as the key value field.

48. (Original) A method according to claim 46, wherein:
defining a second attribute includes defining a third attribute of the first element, the third attribute identifying the second attribute as the key; and
identifying a key value field includes identifying the attribute value field of the second attribute as the key value field.

49. (Original) A method according to claim 43, wherein the schema is an XML schema.

50. (Currently Amended) A computer-readable medium containing a program to add a generic document to a data store, comprising:
software to access a schema for the generic document;
software to locate a key value field in the generic document defined in the schema as a key;
software to load a value from the key value field;
software to store the value in the data store;
software to store the generic document in the data store; and
software to index the generic document in the data store using the value.

51. (Original) A computer-readable medium according to claim 50, wherein the software to identify a key value field includes:
software to locate an element defined in the schema as the key; and
software to select an element value field for the element as the key value field.

52. (Original) A computer-readable medium according to claim 50, wherein the software to identify a key value field includes:
software to locate a first attribute of an element defined in the schema as the key; and
software to select an attribute value field for the first attribute as the key value field.

53. (Currently Amended) A computer-readable medium according to claim 50, wherein the software to store the generic document includes:

software to parse the generic document into objects according to the schema;
software to load values for each object in the generic document; and
software to store the values in the data store.

54. (Currently Amended) A computer-readable medium according to claim 50, wherein:
the generic document is an eXtensible Markup Language (XML) document; and
the schema is an XML schema.

55. (Original) A computer-readable medium containing a program to define a data store, comprising:

software to access a schema;
software to locate an object defined in the schema as a key;
software to define a first data structure in the data store for the object;
software to identify the first data structure in the data store as a key data structure; and
software to define a second data structure in the data store for a generic document
conforming to the schema.

56. (Original) A computer-readable medium according to claim 55, wherein the software to define a second data structure includes:

software to parse the schema into objects; and
software to define a data structure in the data store for each object.

57. (Currently Amended) A computer-readable medium according to claim 55, wherein:
the generic document is an eXtensible Markup Language (XML) document; and
the schema is an XML schema.

58. (Currently Amended) A computer-readable medium containing a program to define a schema for a generic document, comprising:

software to define a first element in the schema, the first element including an element value field; and
software to identify a key value field in the schema to be used as a key in a data store.

59. (Original) A computer-readable medium according to claim 58, wherein:

the software to define a first element includes software to assign a property to the first element as the key; and

the software to identify a key value field includes software to identify the element value field as the key value field.

60. (Original) A computer-readable medium according to claim 58, wherein:
the software to define a first element includes software to define a first attribute of the first element, the first attribute identifying the first element as the key; and
the software to identify a key value field includes software to identify the element value field as the key value field.

61. (Original) A computer-readable medium according to claim 58, wherein the software to define a first element includes software to define a second attribute for the first element, the second attribute including an attribute value field.

62. (Original) A computer-readable medium according to claim 61, wherein:
the software to define a second attribute includes software to assign a property to the second attribute as the key; and
the software to identify a key value field includes software to identify the attribute value field as the key value field.

63. (Original) A computer-readable medium according to claim 61, wherein:
the software to define a second attribute includes software to define a third attribute of the first element, the third attribute identifying the second attribute as the key; and
the software to identify a key value field includes software to identify the attribute value field of the second attribute as the key value field.

64. (Original) A computer-readable medium according to claim 58, wherein the schema is an eXtensible Markup Language (XML) schema.